Article can be accessed online at http://www.publishingindia.com

Multi-view Face Recognition by Neural Network

۲

Sanjay Kumar Dekate*, Anupam Shukla**

the images of the importance. So, we decided to suggest multi-view facial recognition process by morphological features through artificial neural network (ANN). Multiview face recognition, now, is a point of contemplation because in the real world greater than 80% faces in real images are non-frontal. Morphological features are selfdetermining of all the effects like expressions, feelings, illuminations, backgrounds, obstruction, etc. Here, three dissimilar views (front, down, and up) have been taken presented below in the Figure 1.

Figure 1: (a) Front View (b) Down View (c) Up View

۲



Why Artificial Neural Network (ANN)?

The ANN helps when we cannot frame an algorithmic clarification. Like conventional computer uses an algorithm solution i. e. computer follow a set of instruction to resolve certain problem but this doesn't happens with artificial neural network. ANN learns by training; to perform specific task programming is difficult (Abate, Nappi, Riccio & Sabatino, 2007).

Purpose of using Feed-forward Backpropagation Algorithm (FF-BPA)

The reason for using FF-BPA in ANN is its efficacy and dependability. The multi-layered ANN has many limitations like it is restricted to do restricted number of jobs. By back propagation algorithm we are able to

Abstract

()

Multiple views face recognition has become significant in various requisitions, such as observation, human workstation connection, and recreation. A reduction based feature extraction and neural network inspired by biological neurons for learning and recognising the multiple views faces of the person has been presented in this paper. Neural Network (NN) is significant in the places where formulating an algorithmic solution is difficult and we need to retrieve the structure from existing and predefined data. Multi-view face recognition is required here because it's more feasible and reliable than single view face recognition.

Keywords: Multi-views, Facial Recognition, Artificial Neural Network, FF-BPA, Feature Extraction, Segmentation

Introduction

Security of public places from terrorist organisations has become a matter of paramount importance for government. Marker or secret word based approval is common to the point that we can choose the hack of our choice. Biometrics talks of generous elective but they continue of lacunas as well. For example, Iris checking is particularly healthy yet irrationally interfering; finger impressions are publicly avowed, but it's not suitable to non-consentient persons. Obviously, face recognition shows to an unusual argument off between what's publicly agreeable and what's dependable, actually when working under meticulous situations. It's a biometric based credentials or verification of people from digital image or video frame from diverse videos. Faces are recognized by extracting and examining features from

* *Assistant Professor, Government Engineering College, Jagdalpur, Chhattisgarh, India. E-mail: sanjaydekate19814@gmail.com

Volume 3 Issue 1 June 2015

۲

regulate the weights of processing units i. e. neuron for bringing more learning in our neural network. The other idea behind using this procedure is to regulate the error in the hidden layers for minimizing the net error.

Training Function

For training the neural network TRAINSCG function is used for its memory-efficient function. It's used for big dataset and appraises weights and biases by scaled conjugate gradient method. Training of the network stops when any of the subsequent conditions meet:

- Maximum number of iteration has reached.
- Optimum performance is achieved.
- Gradient has reached minimum.
- Maximum time limit exceeded.
- Validation achieved is maximised.

Learning Function

۲

Calculation of the alteration in weight for a specific neuron from the given input and the error uses this learning function. Here, LEARN-GDM is used.

Transfer Function (TANSIG)

Calculation of the output of the layer from its net input is done here. Diverse transfer functions are given below.

Input data is given as a parameter and matrix having size equal to the matrix size of input data and having values between -1 and 1 is returned by TANSIG or the Tan-Sigmoid transfer function. Graph showing the relationship between input and output is given in Figure 2.

Figure 2: TANSIG



Figure 3: LOGSIG



It is Log-Sigmoid transfer function i. e. LOGSIG takes input data as a parameter and returns a matrix having size equal to the matrix size of input data and having values between 0 and 1. Graph showing the relationship between input and output is shown in the Figure 3.

Here, use of logsig function indicates that input data is normalized between 0 and 1 and the demand of output is between 0 and 1.

Regression

It talks about how the actual and the target output are related and linked to each other. Target value is the dependable variable over here and is related to other as independent. Variables whose values are guessed or are not based on result are the dependent variables and the independent variables result is based on actual output. Its classified as below:

Linear

Graph (Figure 4) shows the relationship between target and the independent variable. It's a straight line.

Figure 4: Linear Regression



Non-Linear

Graph (Figure 5) shows relationship between target and predictors and is not linear.

Figure 5: Non-Linear Regression



Multi-view Face Recognition by Neural Network

Multivariate Linear

In this, more than one predictor is used.

Multivariate Non-Linear

In this non-linear regression, more than one predictor is used.

Literature Review

۲

Fiducial points are suggested and implemented morphological method for face recognition (Tiwari, Shukla, Prakash, Sharma, Kumar & Sharma, 2009). Preprocessing, segmentation of the face image, feature mining and then training neural network are the highlights of this system. In preprocessing, edges are pointed by sobel operator and then segmentation has been done. Then, centre of gravity has been cumulated for each segmented region and geometrical features have been taken into account further. The key result achieved is around 70% on testing dataset. To concentrate attributes and utilize probabilistic classifiers to make a various sees face classifier a studying strategy called a recursive nonparametric discriminant dissection has been utilized on the premise of conventional Fisher discriminant investigation and nonparametric discriminant examination (Wang & Ji, 2007; Bressan & Vitria, 2003). The RNDA unfolds Gaussian uncertainties of Fisher discriminant dissection (FDA), and it can deal with more general class disseminations. Its computational intricacy is minimized and thus RNDA improves and better the conventional nonparametric discriminant analysis (NDA). Haar features are preferred precision wise by RNDA as a part of discovering and finding objects of complex shapes. Class circulations are represented by Histograms of extracted properties and to develop probabilistic classifiers. RNDA characteristics are clubbed together with Adaboost to develop a multiview face locator. The plan is linked to both multi-view face and eye identification, and test outcomes show a developed execution over existing strategies. As the three-dimensional geometric relationship is set up and established, the facial aspects are differentiated. The human facial properties have been naturally distinguished focused around three dimensional geometric connections, unfolding what summed up to 67 characteristic points and 24 characteristic lines more than the definitions joined with MPEG-4 (Fang & Fang, 2011). Sensible outcomes have been achieved by thickness point cloud from a non-committed head scanner with 86. 6% validity in the

5 mm range and the 31 characteristic points have been researched on the human face, with 96. 7% validity of short of what 5 mm (Fang & Fang, 2011).

Methodology

Preprocessing

Initially, the interested image to be focused is read into the file. Reducing the dimension of the image is the following step by converting it into gray image. Now a 2-D image is formed. After this, edges in the image are pointed out using Sobel edge detection method. Sobel edge detection is a unique method, which formulates a rough idea of the gradient of image intensity. Either Sobel gives corresponding gradient vector or its normalised value for each pixel of the image. Its computation is based on the convolution of image matrix with a small integer valued filter in horizontal direction and vertical direction. It uses a 3*3 matrix which is shown in equation 1.

$$G_x = -2 \quad 0 \quad +1 \quad +1 \quad +2 \quad +1$$

$$G_x = -2 \quad 0 \quad +2 \quad *A \text{ and } G_y = 0 \quad 0 \quad 0 \quad *A$$

$$-1 \quad 0 \quad +1 \quad -1 \quad -2 \quad -1$$

Equation 1: Sobel mask

۲

15

where, A = image matrix.

The coupled slope mask has lines of high distinction in the picture. These lines don't exactly show the layout of the object of investment. Juxtaposed with the first picture, perforations are seen in the lines circumscribing the item in the gradient mask. The straight perforations dissipate if the Sobel picture is broadened using immediate organising components, which we can make with the ability.

The parallel slope mask is enlarged by the top to down organizing component reproduced by the even organizing. The photo gets expanded by the *imidilate* extent. The dilated slope veil indicates the diagram of the cell, however there are still openings in the inside of the cell. The openings are filled using the *imfill* capacity.

Feature Extraction

After division of face images into different segments, we intensity-weighted centroids of all different parts are computed. The diagram is shown.

Labeling of divided regions is done using *bwlabel* and then using the *regionprops* functions the *PixelidxList* and *PixelList* are to be computed. The values of x and y coordinates of each divided area are found by *PixelList* and *PixelldxList* is used to compute the grayscale values for each pixel in the divided area, and thus the centroid. After centroid of each segment is found, those segments which are of interest are noted (as shown below in the Figurein Figure). After finding the centroids, the distance between two eyes and the nose length are calculated respectively by Euclidean distance and Pythagorean triplet formula.

Other features are also taken in the same way.

$$D = \sqrt[2]{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$
$$a^2 + b^2 = c^2$$

Equation 2: Euclidean distance and Pythagorean triplet formula

Thirteen features of 200 people in three positions (upper, front and lower view) have been taken which are shown in Table 1.

<i>S. N</i> .	Facial Features Extracted
1	Right Eye Height
2	Right Eye Width
3	Right Eye Area
4	Left Eye Height
5	Left Eye Width
6	Left Eye Area
7	Mouth Height
8	Mouth Width
9	Nose Width
10	Face Height
11	Face Width
12	Face Area
13	Center of Mass

Table 1:Facial Features Extracted

The Viola-Jones Face Detector: Methods

Here the principle is to read a sub-window which can recognise faces over a given set of input image. The standard approach is to resize the image to various sizes and then the fixed size detector is run on these images. Volume 3 Issue 1 June 2015

۲

This is a time consuming approach as we calculate the various size images.

On the other hand, Viola-Jones rescales the detector rather than doing it with the input image and runs the detector several times through the image – each time varying the image size. It can be thought that the approaches consume equal time, but Viola-Jones has made a size invariant detector, it requires the same number of calculations irrespective of the size. This detector is constructed by integral image and also using rectangular features reminiscent which is Haar wavelets. This will be discussed in subsequent sections.

The Scale Invariant Detector

Firstly in the Viola-Jones face detection algorithm the input image is transformed to an integral image. We do this by equating every pixel to the overall sum of all pixels above and to the left of that pixel. This is shown in Figure 6.

-gare of - interinge	Figure 6:	The Integral Image
----------------------	-----------	--------------------

1	1	1	1	2	3
1	1	1	2	4	6
1	1	1	3	6	9

Thus we can calculate the sum of all pixels inside any given rectangle from four values. These values represent the pixels in the integral image these pixels overlap with the corners of the rectangle in the input image. This is shown in Figure 7.





It is evident from Figure 7 that Rectangles B and C include Rectangle A thus area of A has to be added to get the required area.

So it is clearly visible that sum of pixels in the arbitrary rectangles can very well be calculated. The Viola-Jones face detector utilizes a given sub window consisting of

Multi-view Face Recognition by Neural Network 17

two or more arbitrary rectangles, which exhibit certain types of features, some of which are shown in Figure 8.

Figure 8: The Different Types of Features

1	1	1		1	2	3
1	1	1		2	4	6
1	1	1		3	6	9
Inp	ut im	age	I	nteg	gral in	nage

From each of the above shown features a single value can be obtained by subtracting the sum of white rectangles from the sum of black ones.

Viola-Jones has empirically found that a detector with a base resolution of 24*24 pixels gives satisfactory results. As depicted in Table 1, a total of 160000 features can be obtained on allowing all shapes and sizes. This number thus outnumbers the 576 features that were obtained



۲







previously using binary PSO-NN.

The flow chart of binary PSO-NN is shown in Figure 9.

The local optimum solution is searched using FFBP algorithm. The FF-BPA uses following steps to learn:

- In a training set, each input pattern is given to the input unit and then propagated forward.
- The error is calculated when the pattern arrives the outer layer by comparing it with the actual output unit.
- The errors so calculated for every output pattern are thus propagated back form output to input to maintain the balance of weight for each pattern.
- After the back-propagation being able to learn proper or right calculations, this is then tested for the other remaining signals.

The working of the BP algorithm is shown in Figure 10.

The weight factor for the hidden-layer neurons front input layer is calculated using the formula

$$W_{ij}^{(k+1)} = W_{ij}^k - \gamma \frac{\delta E^{(k)}}{\delta W_{ij}}$$
 Equation 3: Weight Factor

۲

Here k denotes the number of iteration, I denotes index of input neuron, j denotes index of hidden neurons. Error is calculated using the formula

$$E = \frac{1}{2} \int_{l=1}^{p} (t_l - o_l)^2$$
 Equation 4: Error

Here, p denotes the number of output neurons, l denotes index of neurons, t1 and O1 are the target and actual outputs. The activation, net function and output are calculated using the formulae given below.

$$S_{i} = \frac{1}{1 + e^{(-\delta n e t_{i})}}$$
$$net_{i} = \prod_{l=1}^{n} W_{ll} x_{l} + W_{ln+1}$$
$$O_{i} = \prod_{l=1}^{m} V_{ll} S_{l} + V_{ln+1}$$

Equation 5: Activation, Net Function and Output

Here n denotes number of input neurons, m denotes num of output neurons. FF-BPA is shown using the flow diagram in Figure 11.

International Journal of Knowledge Based Computer Systems

Figure 11: FF-BPA Flow Chart

The features used by the neural network have been shown in Table 2.

Upper, front and lower all the three unlike views have been taken into account. Now there are 50 samples having nine illustration of each. Then, some distinct attributes, Volume 3 Issue 1 June 2015

۲

Type of Network	FF-BPA
Learning Function	LEARNGDM
Function for Performance	MSE
Number of total Layers	3
Input layer neurons	14
Hidden layer neurons	25
Function for Transfer	LOGSIG
Function for Training	TRAINSCG

Table 2:Parameters for NN

let's say six from each image are extracted. 300 rows and 14 columns form the dataset.

Attributes from the images have been pulled out in order to allow dataset to give as the input to ANN to learn. The next operation will be performed with the help of extracted features which are saved in excel. Once all the attributes are extracted, the maximum value of each attribute is calculated followed by all values of each attribute which

Table 3:Extracted Feature Values

0.595745 0.595745 0.354912 0.568182 0.568182 0.322831 0.527778 0.534884 0.780488 0.748466 0.748466 0.560202 0.969263 1 0.595745 0.595745 0.354912 0.568182 0.568182 0.322831 0.527778 0.534884 0.780488 0.748466 0.748466 0.560202 0.969263 1 0.553191 0.553191 0.306021 0.545455 0.545455 0.297521 0.444444 0.44186 0.756098 0.711656 0.711656 0.506455 0.969263 1 0.553191 0.553191 0.306021 0.545455 0.545455 0.297521 0.444444 0.44186 0.756098 0.711656 0.711656 0.506455 0.969263 1 0.617021 0.617021 0.380715 0.590909 0.590909 0.349174 0.444444 0.44186 0.829268 0.730061 0.730061 0.53299 0.969263 1 0.617021 0.617021 0.380715 0.590909 0.590909 0.349174 0.444444 0.44186 0.829268 0.730061 0.730061 0.53299 0.969263 0.638298 0.638298 0.407424 0.590909 0.590909 0.349174 0.611111 0.604651 0.682927 0.699387 0.699387 0.489141 0.976064 0.943696 0.638298 0.638298 0.407424 0.590909 0.590909 0.349174 0.611111 0.604651 0.682927 0.699387 0.699387 0.489141 0.976064 0.943696 0.595745 0.595745 0.354912 0.613636 0.613636 0.37655 0.611111 0.604651 0.634146 0.613497 0.613497 0.376378 0.976064 0.943696 0.595745 0.595745 0.354912 0.613636 0.613636 0.37655 0.611111 0.604651 0.634146 0.613497 0.613497 0.376378 0.976064 0.943696 0.595745 0.595745 0.354912 0.613636 0.613636 0.37655 0.444444 0.44186 0.731707 0.760736 0.760736 0.57872 0.976064 0.943696 0.595745 0.595745 0.354912 0.613636 0.613636 0.37655 0.444444 0.44186 0.731707 0.760736 0.760736 0.57872 0.976064 0.943696 0.680851 0.680851 0.463558 0.704545 0.704545 0.496384 0.666667 0.674419 0.878049 0.834356 0.834356 0.69615 0.969144 0.949266 0.680851 0.680851 0.463558 0.704545 0.704545 0.496384 0.6666667 0.674419 0.878049 0.834356 0.834356 0.69615 0.969144 0.949266 0.744681 0.744681 0.55455 0.704545 0.704545 0.496384 0.805556 0.813953 0.756098 0.797546 0.797546 0.63608 0.969144 0.949266 0.744681 0.744681 0.55455 0.704545 0.704545 0.496384 0.805556 0.813953 0.756098 0.797546 0.797546 0.63608 0.969144 0.949266 0.468085 0.468085 0.219104 0.5 0.5 0.25 0.486111 0.488372 0.878049 0.717791 0.717791 0.515225 0.969144 0.949266 0.468085 0.468085 0.219104 0.5 0.5 0.25 0.486111 0.488372 0.878049 0.717791 0.717791 0.515225 0.969144 0.949266 0.702128 0.702128 0.492983 0.795455 0.795455 0.632748 0.638889 0.627907 0.853659 0.840491 0.840491 0.706425 0.970823 0.983385 0.702128 0.702128 0.492983 0.795455 0.795455 0.632748 0.638889 0.627907 0.853659 0.840491 0.840491 0.706425 0.970823 0.983385 0.787234 0.787234 0.619737 0.795455 0.795455 0.632748 0.833333 0.837209 0.853659 0.797546 0.797546 0.63608 0.970823 0.983385 0.787234 0.787234 0.619737 0.795455 0.795455 0.632748 0.833333 0.837209 0.853659 0.797546 0.797546 0.63608 0.970823 0.983385 0.468085 0.468085 0.219104 0.545455 0.545455 0.297521 0.583333 0.581395 0.804878 0.742331 0.742331 0.551056 0.970823 0.983385 0.468085 0.468085 0.219104 0.545455 0.545455 0.297521 0.583333 0.581395 0.804878 0.742331 0.742331 0.551056 0.970823 0.983385 0.595745 0.595745 0.354912 0.590909 0.590909 0.349174 0.847222 0.860465 0.731707 0.803681 0.803681 0.645903 0.957108 0.913969 0.595745 0.595745 0.354912 0.590909 0.590909 0.349174 0.847222 0.860465 0.731707 0.803681 0.803681 0.645903 0.957108 0.913969 0.638298 0.638298 0.407424 0.704545 0.704545 0.496384 0.736111 0.744186 0.731707 0.680982 0.680982 0.463736 0.957108 0.913969 0.638298 0.638298 0.407424 0.704545 0.704545 0.496384 0.736111 0.744186 0.731707 0.680982 0.680982 0.463736 0.957108 0.913969 0.489362 0.489362 0.239475 0.477273 0.477273 0.227789 0.819444 0.837209 0.902439 0.638037 0.638037 0.407091 0.957108 0.913969 0.489362 0.489362 0.239475 0.477273 0.477273 0.227789 0.819444 0.837209 0.902439 0.638037 0.638037 0.407091 0.957108 0.913969 0.765957 0.765957 0.586691 0.863636 0.863636 0.745868 0.944444 0.953488 0.804878 0.846626 0.846626 0.716775 0.966494 0.893033



۲

Multi-view Face Recognition by Neural Network 19

۲

are further divided by the maximum value of respective attribute. It brings all the values between 0 and 1 and this is called normalisation. Features in excel have been shown in Table 3.

The extracted dataset has been divided into following three categories:

Training Dataset

۲

The weight of neural dataset is adjusted in order to control and minimize the error of the data by training dataset also used for learning the neural network. Here, a large percentage of the dataset 60% has been used for training the neural network. This dataset serves as input of the NN tool.

Testing Dataset

The features of the person used for training the neural network are taken into account in this dataset, majorly used to cross-examine the outcomes of the trained network and to confirm the predictive power of the network. Here, 40 percent of the dataset has been used and makes us sure if our model is able to recognize characters or not

۲

Table 4:Target Output

10 M	0 8	0 0	0	1	0 0	0	0 0	. 0		0	0	1 0	8 0		0 0		0 0	0	1 0	- 1	0	1. 1	0 0	0	0. 1	0 0
	2 2			-			-			2								-		-9-		1			1	
1 0	0 4	0 0	<u> </u>		0.0	. 0	V 0	. 0		V 1	0	1 0	1 0		8 6		0 0	- V	1 0		0	E	0 0	4	V	0 0
5 0	0 0	0 8	0	1	0 0	0	0 0	0	0	0 1	1 0	1 0	8 0	8 20 3	0 0	0	0 0	0	1 0	1	ů.	1.	0 0	0	0	0 0
4 4	0 4	0 4	0		A 8		£ 0			A	1 0				0 0		6 6	0	1 0		0		0.0			0 0
2 (V)	0 0	0 1	V	1	V 1	0	6 9			V 1	1 1	1 V	1 V		0 0		0 0		1 V		ų	- C	V 1		V	4 0
1 0	0 8	0 1	0	1	0 8	0	0. 0	0	0	0	1 0	8 0	1 0	1. 1. 1.	8 8	- 0	0 0	0	0	- 1	0	- E	0 0	0	0	8 0
× A	0 0	0 8			0 E	0	E 4			0	0				0 0	100	0 4	6	a 0.			# S	a			4 6
8 4	0 8	0 8	0		0 0		E 9			V 1	8 0	0	8 V		0 0		0 0	- 0	0		0	- E.	0 8	0	V	9 0
0 1	0 0	0 0	0	1	0 0	0	0 0	0		0 1	1 0	1 0	8 0		0 0	- 0 - 1	0 0	0	1 0	- 1	0	E	0 1	0	0 1	0 0
N 4	6 8	0.4	0		0. 8		A 4			6 1	0		1 0		0 0		A A		1 0		6		A 8	A		0 0
V 1	V. 9	V 8	V		Q E		V	V		V 1	· ·	8 V	* V		V V.		V V	V	• v				V 1	V.	- N.	4 V
0 1	0 0	0 1	0	1	0 1	0	0 0	0	. 0	0	1 0	1 0	0 0	2 1 5	0 0	- 0	0 0	0	0	- 1	0	1.	0 0	0	- R - 9	0 0
0.1	0 8	0 4	0	1	0.8	.0	6 0		4	6 1	0	1 0	1 0		0 0	4	0 0	6	1 0		0	1.	A 8	4		0 0
- 8 10	8 0			-	V		P				• v.	* v	* v		0 0		V 9		8 V.	_			×		8	4 0
0 1	0 0	0 0	0	1	0 0	0	0 0	0		0 1	8 0	1 0	8 0	- R - 1	0 0		0 0	0	1 0	- 1	0	10 3	0 0	0	0 1	0 0
0 1	0 0	0 8	ń.	1	0 8	0	6 0	0	4	ô	0	1 0	8 0	8 SM 8	0 0	4	0 8	n.	1 0	1	0	B	0 1	0	0	0 0
	- X				× *						* V				× ×		v v		* *	_						* *
0 0	1 4	0 8	0	1	0 0	0	0 0	0	0	0	1 0	8 0	1 0	6 3 0 6	0 0		0 0	0	1 0	- 1	0	- E (0 0	0	0	0 0
0 0	1 8	0 1	0	1	0 E	.0	0 0	0		6	1 0 :	1 0	E 0		0 0		0 0	.0	1 0		0	B	0.6	0	0 1	0 0
					1 1		-								<u></u>	_	-		1 2			-	2-0			
0 0	5 4	0 0	0		0 0	0	0 0	0	Q	0	I 0	0	0 0	- 1 - S	0 0	0	0 0	0	1 0	1	0	- R 1	0 0	0	0	0 0
0 0	1 4	0 0	0	1	0 8	0	0 0	0		0 1	1 0	1 0	1 0		0 0		0 0	0	1 0	1	0	1.	0 0	0	0 1	0 0
	1				A			1					1			2			1 .							
.0 .0	- (B)	0 0	V		V 0	0	0 0	. 0	4	0.1	1 V	1 0	1 0	- 18 - V	V 0		V 0	U	1 U		. 0	10 C	v t	- U -	V	0 0
0 0	1 0	0 1	0	1	0 8	0	0 0	0	0	0 1	0 1	1 0	1 0	1	0 0	0	0 0	0	0	1	0	8	0 0	0	0	0 0
0 0	0 1	0.4	0		0 0	0	8 4	A		0	1 0	1 0	1 A		0 0	4	0 4	6	4 6		0		0		6	0 0
0 0	V . 1	0 1	v		V 8	4	e 4			4	, V	4 0	. 0		4 6		V 4	V	4 4		0		v 1	4		* 0
0 0	0 1	0 1	0	1	0 0	0	0 0	0	0	0	0 8	0	1 0	1 1	0 0	0	0 8	0	1 0	1	0	10.0	0 0	0	0 1	0 0
0 0	0 +	0 4	0	1	0 5	0	0 0		4	0	0 1	1 0	1 0		0 0		0 4	0	1 0		0	1	0 0		0	0 0
	× I	0 1	<u>v</u>		v 8		6 9					4 0		1 1	× 8.		v 0		a 4		V	- C	y 1		V	
0 0	0 1	0 1	0	1	0 0	0	0 0	.0	0	0 1	0	1 0	0		0 0	1	0 0	0	0		0	1. 1	0 0	0	Q	0 0
0 0	0 1	0 1	Ô	1	0 B	ő	0 0	6	0	0 0	0 1	1 0	1 0		0 0	4	0 4	Ô.	1 0	1	Ó	1	0 1	0	6	0 0
	1 1			-	2 2		1 1			2 1					· ·							-			-2	
0 0	0 1	0 0	Q	1	0 0	0	0.0	- 0	- 4	0 1	1 0	1 0	1 0		0 0	- 0	0 0	0	0	- 1	0	- P	0 0	0	0	0 0
6 6	0 8	1 1	0	1	0 1	. 0	0.0	. 0	. 4	0 1	0 1	1 0	8 0	1	0 0	- 1	0 0	0	1 0	- 1	. 0	1	0 0	0	0 1	0 0
				-	A																					
6 9	0 0	1 8	U	1	0 0	0	E 9	. 0		0 1	8 0	0 0	- P - U	6 (R)	0 0	9	0 0	0	1 V.	- 10	0	- B(< -)	9 8	0	- P	8 0
0 0	0 1	1 1	0	1	0 0	.0	0 0	. 0		0 1	1 0	1 0	8 0	10	0 0	1	0 0	0	1 0		0	1. 3	0 0	0	0 1	0 0
6 4	0 0	4 4			0.1		6 6			6	1 0				0 0		6 8		1 0				a			0 0
	0 4				0 1		6 4	- · · · ·		.0	, V.	8 V	6 Q		0 0		0 0	V	9 V.		.0		0 1	- V.	V	4 0
0 0	0 1	1 4	- Ő	1	0 8	0	0 0	0	0	0 0	1 0	1 0	1 0		0 0	0	0 0	0	0	- 1	0	E	0 0	0	0	0 0
6 6	6 4	1 1	0		0.0		6 0		4	6 1	1 0	1 0	E 0		0 6	A	6 8	6	4 0		6		A	4	6 .	a 6
0 0	0 0	1 1	Ó	1	0 0	Ó	0 0	0	0	0 1	1 0	1 0	1 0	. 0	0 0	0	0 0	0	1 0	1	0	1	0 0	0	0	0 0
0 0	0 0	1 1	0	1	0 0	0	0 0	0	0	0	1 0	1 0 1 0	1 0 1 0	8	0 0	0	0 0	0	1 0	1	0	8	0 0 0 6	0	0	0 0 0 0
0 0	0 0	1 0	0	1	0 0	0	0 0	0	0	0		1 0 1 0	8 0 8 0	8	0 0	0	0 0	0		1	0	8. 8.	0 0	0	0	0 0 0 0 2 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 1 0 1	0 0	1	0 0 0 0 0 0	0 0 0	0 0	0 0 0	0 0 0	0 1		1 0 1 0 1 0	8 0 8 0 8 0	0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0		1	0 0 0	8 8	0 0 0 8 0 0	0	0	0 0 0 0 0 0
0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 II 0 1 0 1	0 0 0 0 0 0	1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 0 0	0 0 0	0 0 0 0	0 1	1 0 1 0 1 0	1 0 1 0 1 0	8 0 8 0 8 0	8 8 8	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 0 3 0 0	0 0 0		1	0 0 0 0	8 8 8	0 0 0 0 0 0 0 0		0 0	0 0 0 0 0 0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 1 0 1 0 1	0 0 0 0 0 0	1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 1		1 0 1 0 1 0 1 0	8 0 8 0 8 0 8 0 8 0	8 8 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0		1	0 0 0 0	8 8 8	0 0 0 0 0 0 0 0		0	0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 1 0 1 0 1 0 1	0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0		1 0 1 0 1 0 1 0 1 0 1 0	0 8 0 8 0 8 0 8 0 8	8 8 8 8	0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0		1	0 0 0 0	8 8 8 9	0 0 0 0 0 0 0 0	0	0	a 0 a 0 a 0 a 0
0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 8 0 1 0 1 0 1 0 1 0 1	0 0 0 0 0	1 1 1 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0			1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 8 8 8 8	0 0 0 0 0 0 0 0 0 0 0 0	0 9 0 0 0 0	0 0 0 9 0 8 0 0 0 0 0 0 0 0	0 0 0 0	1 0 1 0 1 0 1 0 1 0 1 0 1 0	1	0 0 0 0 0	8 8 8 9 8 8	0 0 0 0 0 0 0 0 0 0 0 0	000000000000000000000000000000000000000	0 0 0 0 0	0 0 8 0 8 0 8 0 8 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 8 0 8 0 8 0 8 0 8 0 8 0 8	1 II 0 1 0 1 0 1 0 1 0 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 1	0 8 0 8 0 8 0 8 0 8 0 8 0 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0				8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 8 8 8 8	0 0 9 6 9 6 0 6 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		1	0 0 0 0 0 0	8 6 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0		0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 II 0 1 0 1 0 1 0 1 0 1 0 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 1	0 8 0 8 0 8 0 8 0 8 0 8 0 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 0 8 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		1 1 1 1 1	0 0 0 0 0 0	E E E E E E E E	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		000000000000000000000000000000000000000	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 II 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1	0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0			8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 9 0 0 0 0 0 0 0	0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	8 0 9 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 9 0	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0	8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	0 0 0 0 0 0 0 0 0 1	1 1 1 1 1 1 1 1 1	0 B 0 B 0 B 0 B 0 B 0 B 0 B 0 B 0 B	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0			8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	6 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 9 0 0 0 0 0 0 0 0 0	0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0 0 0	8 8 8 8 8 8 8 8 8	3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0		0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8	1 8 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	0 0 0 0 0 0 0 1 1	1 1 1 1 1 1 1 1 1	0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	6 6 6 6 6 6 6 6 6	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 9 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0	8 8 8 8 8 8 8 8 8 8 8 8	0 0 3 0 3 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0		000000000000000000000000000000000000000	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8	1 8 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 8 0 8	0 0 0 0 0 0 0 1 1 1		0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1		8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	6 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 9 0 9 0 9 0 0 0 0 0 0 0 9 0 9 0 9	0 0 0 0 0 0 0 0 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0		0 0 0 0 0 0 0 0 0 0 0	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 8 0 9 0 9 0 8 0 8 0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9	1 8 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 8 0 8 0 8 0 8 0 8 0 8 0 8	0 0 0 0 0 0 0 1 1 1 1		0 B 0 B 0 B 0 B 0 B 0 B 0 B 0 B 0 B 0 B		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	6 6 6 6 6 6 6 6 6 6 7 6 7 7 7 7 7 7 7 7	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 9 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0	E E E E E E E E E E E E E E E E E E E	0 0 3 0 3 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
			0 0 0 0 0 1 1 1 1		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8			0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					8 0 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0	E E E E E E E E E E E E E E E E E E E	0 0		0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0	0 0 0 0 0 0 1 1 1 1 1		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1		1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0		8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8		0 0 0 0 0 1 1 1 1 1 1 1		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 9 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	3 0 0 3 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9		0 0 0 0 0 0 1 1 1 1 1 1 1 0					0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0		8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	3 0 0 3 0 0 0 3 0 0 0 3 0 0 0 3 0 0 0 3 0 0 0 3 0 0 0 3 0 0 0 3 0 0 0 3 0 0 0 4 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 9 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	3 0 0 3 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9	1 8 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0	0 0 0 0 0 0 1 1 1 1 1 1 1 1 0		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 1 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0	0 0 8 0 8 0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0		8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 8 0 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8	1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	0 0 0 0 0 1 1 1 1 1 1 1 0 0	5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1	0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 8 0 8 0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 8 0 8 0 8 0 8 0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9		0 0 0 0 0 1 1 1 1 1 1 0 0 0		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0		8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 9 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	6 6 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0
6 0 6 0 6 0 6 0 6 0 6 0 6 0 6 0	0 8 0 8 0 9 0 0	1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	0 0 0 0 0 1 1 1 1 1 1 0 0 0		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8			000000000000000000000000000000000000000		0		8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0		8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 9	1 8 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0	5 5 5 5 5 5 5 5 5 5 1 1 1 1	0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 8 0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0	I O I O	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 3 0 2 0 3	1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 8 0 8 0 8 0 8 0 9 0	8 0 8 0		8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				E E E E E E E E E E E E E E E E E E	3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 3 0 3		0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8							8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0		8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	3 0 0 3 0 0 0 3 0 0 0 0 3 0 0 0 0 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0
0 0 0 0	0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3	1 0 0 1	0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0	5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 8 0 8 0 8 0 9 0	I O I O	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	E F F F F F F F F F F F F F	1 0 0 0 0 1 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0
	0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 3 3 0 3 3 0 3 3 0 3 3 0 3 3 0 3 3 0 3 3		0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 9 0 9	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	E E E E E E E E E E E E E E E E E E E	3 0 0 3 0 0 0 3 0 0 0 0 3 0 0 0 0 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0
	0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9		0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 9 0 9 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0		6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0		0 9 0 9						1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0
	0 3 0 3		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 9 0 9					E 5 5 5 5 5 5 5 5 5 5 5 5 5	3 0 0 0 3 0 0 0 0 3 0 0 0 0 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0	0 3 0 0 0 3 0 0		0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0		0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0		6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0		0 9 0 9				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	E 5 6 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0
	0 8 0 8	1 8 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1			0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8							8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0		0 9 0 9					E E E E E E E E E E E E E E	3 0 0 3 0 0 0 3 0 0 0 0 3 0 0 0 0 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
	0 8 0 8		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 8 0 8 0 0 1 0 1 0				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0	8 0 9 0 9 0 9 0 9 0		6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0		0 0 0 0		1 0 3 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 9 0 1 0				3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 4 0 0 5 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
	0 8 0 8	1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 1 8 1 8 1 8				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 8 0 8 0 8 0 9 0	8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0		1 0 3 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0			E E E E E E E E E E E E E E	3 0 0 3 0 0 3 0 0 0 3 0 0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0	0 8 0 8				0 8 1 8 1 8 1 8			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0		0 0		1 0 1 0				3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0 3 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
	0 8 0 8				0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 1 8 1 8 1 8				8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9			8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0	8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0		0 0 0 0 0 0		0 0 0 8 0 0		1 0 1 0				1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0	0 3 0 3		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1				8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9			8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0			0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0		1 0 1 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
		1 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 1 8 1 8 1 8 0 1			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0			0 0		0 0		1 0 1 0				3 0 0 3 0 0 0 3 0 0 0 0 3 0 0 0 0 0 3 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

20 International Journal of Knowledge Based Computer Systems

Volume 3 Issue 1 June 2015

۲

Validation Dataset

This dataset is used to reduce the over-fitting and the network is unknown to this dataset. Accuracy of the trained network is measured with the help of this dataset. The condition of over-fitting the network is when the precision for the dataset of training increases but the precision of validation dataset remains constant.

Target Dataset

In each column of the target dataset, values corresponding to each person instance are taken as 1; rest all values are kept 0. It is prepared to set target for the Neural Network to learn. The prepared target dataset is shown in Table 4.

Experiment Results

۲

The experiment has been conducted on a sample of 100 individuals. Each individual has three views and each view has 5 samples. Their images have been taken under constant light so as to ensure that their all five views look alike. The dataset has been divided into 60%-40% in which 60% dataset has been used for training and 40% dataset has been used for testing the trained PSONN. The similarity in face images helped in better edge detection and segmentation process leading to smooth extraction of different features of every particular individual. This constant light effect helped in achieving 95% accuracy on testing dataset and 93% on training dataset.

Training Performance

Regression tells how the actual output is close to the target output. It depicts the relationship between the target values being the dependable variable and the one or more than one independent variable. The dependent variables are those variables whose values are to be predicted and the independent variables are those variables which form the basis of prediction results. The regression graph obtained in the training phase is shown in Figure 12.

Testing Performance

Figure 12: Regression Graph













Figure 15: Regression Graph for Testing

Mean square error is computed by the performance graph in each epoch while training the neural network. Mean of the addition of the square of the errors obtained by subtracting the actual output from the target output for each neuron.

Conclusions

۲

Inference drawn from the experiment of *Tiwari et al.* (2009) on Grimace face database using morphological feature method is that due to robustness in the light effect, was somewhere in between 60% and 70% and they performed only on front view. In this paper, three views are considered, up view, down view and front view, and all the images have been taken under constant light effect. This constant light effect improved the final outcome and performance has reached to 91% for training dataset and 91% for testing dataset.

Limitation and Future Work

Facial recognition is becoming difficult job and thus facial recognition system should be independent of any type of constraints. In this paper, Independence of pose, expression, orientation and presence of accessories is deeply and solely are taken into considerations. But, there is a drawback that the face images have been taken under constant light effect. Under variable light effect, result may not be the same for sure. Also, only digital images are taken into considerations, which can be extended for video-based face recognition as it has an upper hand over still digital images.

References

- Abate, A. F., Nappi, M., Riccio, D., & Sabatino, G. (2007).
 2D and 3D face recognition: A survey. *Pattern Recognition Letters, October,* 28(14), 1885-1906.
- Balasubramanian, M., Palanivel, S., & Ramalingam, V. (2009). Real time-face and mouth recognition using radial basis function neural networks. *Expert Systems with Applications, April*, 36(3), 6879-6888.
- Bressan, M., & Vitria, J. (2003). Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15), 2743-2749.
- Fang, S. Y., & Fang, J. J. (2011). Automatic head and facial feature extraction based on geometry variations. *Computer-Aided Design, December*, 43(12), 1729-1739.
- Gonzalez-Ortega, D., Diaz-Pernas, F. J., Martinez-Zarzuela, M., Anton- Rodriguez, M., Diez-Higuera, J. F., & Boto-Giralda, D. (2010). Real-time hands, face and facial features detection and tracking: Application to cognitive rehabilitation tests monitoring. *Journal of Network and Computer Applications*, July, 33(4), 447-466.

۲

- Gudise, V. G., & Venayagamoorthy, G. K. (2003). Comparison of particle swarm optimization and back propagation as training algorithms for neural networks. Proceedings of the 2003 IEEE Conference on Swarm Intelligence Symposium, April, 19 (25), 110-117.
- Jain, A. K., & Park, U. (2009). Facial marks: Soft biometric for face recognition. 16th IEEE International Conference on Image Processing, November, 7(17), 37-40.
- Llonch, R. S., Kokiopoulou, E., Tosic, I., & Frossard, P. (2010). 3D face recognition with sparse spherical representations. *Pattern Recognition*, March, 43(3), 824-834.
- Mohammed, A. A., Minhas, R., Wu, Q. M. J., & Sid-Ahmed, M. A. (2011). Human face recognition based on multidimensional PCA and extreme learning machine. *Pattern Recognition*, October-November, 44(10-11), 2588-2597.

۲

۲

22 International Journal of Knowledge Based Computer Systems

Volume 3 Issue 1 June 2015

۲

- Tiwari, R., Shukla, A., Prakash, C., Sharma, D., Kumar, R., & Sharma, S. (2009). *Face recognition using morphological method*. IEEE International Conference on Advance Computing, March, 1(7), 529-534.
- Wang, P., & Ji, Q. (2007). Multi-view face and eye detection using discriminant features. *Computer Vision and Image Understanding, February*, 105(2), 99-111.

۲

- Yun, T., & Guan, L. (2009). Automatic fiducial points detection for facial expressions using scale invariant feature. IEEE International Workshop on Multimedia Signal Processing, October, 11(9), 1-6.
- Zhao, M., Li, P., & Liu, Z. (2008). Face recognition based on wavelet transform weighted modular PCA. Proceedings of the Congress in Image and Signal Processing, 589-593.